Software in everything
Algorithms in everything … in Engineering and Science
Using MATLAB & Simulink to Build Algorithms in Everything

Simplifying your work…

…often at higher levels of abstraction.
Using MATLAB & Simulink to Build Algorithms in Everything

Inputs → Design → Outputs
Artificial Intelligence

*The capability of a machine to match or exceed intelligent human behavior by training a machine to learn the desired behavior*
There are two ways to get a computer to do what you want

Traditional Programming

- Data
- Program
- COMPUTER
- Output
There are two ways to get a computer to do what you want

Machine Learning

Data

Output

Model
Artificial Intelligence

Data → Machine Learning → Deep Learning → Model
Using MATLAB and Simulink to Build Deep Learning Models
Using Apps for Ground Truth Labeling
Image and Video Data
Using Apps for Ground Truth Labeling
Signal Data
Using Apps for Ground Truth Labeling
Audio Data
Using Apps for Designing Deep Learning Networks
Scaling Computation for Training Deep Learning
Using Transfer Learning with Pre-trained Models

- VGG-16
- GoogLeNet
- Inception-v3
- DenseNet-201
- Xception
- NasNetLarge
- AlexNet
- Inception-ResNet-v2
- MobileNet-v2
- NasNetMobile
- VGG-19
- ResNet-50
- ResNet-101
- ResNet-18
- Places365-GoogLeNet
- ShuffleNet
- SqueezeNet

Years:
- 2016
- 2017
- 2018
- 2019
Using Models from Other Frameworks

MATLAB ➔ ONNX ➔... ➔ Core ML ➔ MXNet ➔ Caffe2 ➔ PyTorch ➔... ➔ Caffe

Keras-Tensorflow ➔ MATLAB ➔ Caffe
Deploying Deep Learning Applications

- Pre-processing
- Deep Learning Application
- Post-processing

Coder Products

- Intel MKL-DNN Library
- NVIDIA TensorRT & cuDNN Libraries
- ARM Compute Library

MATLAB Coder
GPU Coder
Using MATLAB and Simulink for Reinforcement Learning

Inputs

Data

Design

Machine Learning

Deep Learning

Model

Outputs

MATLAB & SIMULINK

Reinforcement Learning Toolbox

© 2019 The MathWorks, Inc.
Using MATLAB and Simulink for Reinforcement Learning
Using MATLAB and Simulink for Reinforcement Learning

- Data (Inputs)
- Machine Learning
- Deep Learning
- Model (Outputs)

MathWorks Reinforcement Learning Toolbox
Using MATLAB and Simulink for Reinforcement Learning

**Generate Data**
- Scenario Design
- Simulation-based data generation

**Design**
- Machine Learning
  - Deep Learning

**Outputs**
- Model

Inputs

MATLAB & SIMULINK

Simulink
Reinforcement Learning Toolbox
Using MATLAB and Simulink for Reinforcement Learning
Find out more: Deep Learning e Reinforcement Learning per l’intelligenza artificiale

Giuseppe Ridinò
Traccia A – 13:30
Using MATLAB & Simulink to Build Algorithms in Everything

Inputs

Design

Outputs

MATLAB & SIMULINK

© 2019 The MathWorks, Inc.
## Working with Text Data

<table>
<thead>
<tr>
<th>Dept, JobDate, jobno, Vehicleid, UnitNo, Reason, Notes, CostParts, CostLabor, CostTotal</th>
</tr>
</thead>
<tbody>
<tr>
<td>1020, 01/06/2015 12:00:00 AM, 14073, 118743, 14, 04</td>
</tr>
<tr>
<td>1020, 01/14/2015 12:00:00 AM, 14232, 285079, 13, 08</td>
</tr>
<tr>
<td>2111, 01/02/2015 12:00:00 AM, 14006, 1243, 116, 04</td>
</tr>
<tr>
<td>2111, 01/02/2015 12:00:00 AM, 14140, B39109, 178, 04</td>
</tr>
<tr>
<td>2111, 01/03/2015 12:00:00 AM, 14163, 574950, 215, 13</td>
</tr>
<tr>
<td>2111, 01/09/2014 12:00:00 AM, 14165, A00413, 283, 04</td>
</tr>
<tr>
<td>2111, 01/06/2015 12:00:00 AM, 14000, 766153, 248, 08</td>
</tr>
<tr>
<td>2111, 01/06/2015 12:00:00 AM, 14155, 525670, 232, 04</td>
</tr>
<tr>
<td>2111, 01/06/2015 12:00:00 AM, 14157, 621909, 213, 40</td>
</tr>
<tr>
<td>2111, 01/06/2015 12:00:00 AM, 14164, 1226, 117, 13</td>
</tr>
<tr>
<td>2111, 01/06/2015 12:00:00 AM, 14165, 525999, 114, 04</td>
</tr>
<tr>
<td>2111, 01/06/2015 12:00:00 AM, 14172, B34632, 276, 10</td>
</tr>
<tr>
<td>2111, 01/06/2015 12:00:00 AM, 14174, 1469, 122, 10</td>
</tr>
<tr>
<td>2111, 01/06/2015 12:00:00 AM, 14176, 6893, 147, 10</td>
</tr>
<tr>
<td>2111, 01/06/2015 12:00:00 AM, 14176, 6893, 148, 10</td>
</tr>
<tr>
<td>2111, 01/06/2015 12:00:00 AM, 14177, 621907, 208, 10</td>
</tr>
<tr>
<td>2111, 01/06/2015 12:00:00 AM, 14181, 337657, 218, 04</td>
</tr>
<tr>
<td>2111, 01/06/2015 12:00:00 AM, 14182, D-1520, 164, 10</td>
</tr>
<tr>
<td>2111, 01/06/2015 12:00:00 AM, 14183, 525998, 217, 10</td>
</tr>
<tr>
<td>2111, 01/06/2015 12:00:00 AM, 14184, 526000, 225, 10</td>
</tr>
<tr>
<td>2111, 01/06/2015 12:00:00 AM, 14185, 621921, 214, 04</td>
</tr>
<tr>
<td>2111, 01/07/2015 12:00:00 AM, 14188, 004169, 201, 04</td>
</tr>
<tr>
<td>2111, 01/07/2015 12:00:00 AM, 14190, 337656, 219, 04</td>
</tr>
<tr>
<td>2111, 01/07/2015 12:00:00 AM, 14191, B34632, 276, 10</td>
</tr>
<tr>
<td>2111, 01/07/2015 12:00:00 AM, 14196, 1222, 118, 04</td>
</tr>
<tr>
<td>2111, 01/07/2015 12:00:00 AM, 14199, 525655, 626, 04</td>
</tr>
<tr>
<td>2111, 01/09/2015 12:00:00 AM, 14107, 1467, 121, 08</td>
</tr>
</tbody>
</table>
### Working with Text Data

```matlab
% Read table from file
filename = 'example.txt';
t = readtable(filename,'TextType','string');
disp(t(1:20,6:7));
```

<table>
<thead>
<tr>
<th>Reason</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;04 DRIVER'S REPORT&quot;</td>
<td>&quot;PM SERVICE, CHECK TURN SIGNAL, CLUNKING NOISE WHEN DRIVING&quot;</td>
</tr>
<tr>
<td>&quot;08 PM SERVICE&quot;</td>
<td>&quot;SERVICER08,EXT,5604&quot;</td>
</tr>
<tr>
<td>&quot;04 DRIVER'S REPORT&quot;</td>
<td>&quot;NEED 4 PLOW PINS&quot;</td>
</tr>
<tr>
<td>&quot;04 DRIVER'S REPORT&quot;</td>
<td>&quot;INSTALL SPINNER ASSY&quot;</td>
</tr>
<tr>
<td>&quot;13 SNOW BREAKDOWN&quot;</td>
<td>&quot;DON'T START&quot;</td>
</tr>
<tr>
<td>&quot;04 DRIVER'S REPORT&quot;</td>
<td>&quot;DOG BONE PIN BROKEN&quot;</td>
</tr>
<tr>
<td>&quot;08 PM SERVICE&quot;</td>
<td>&quot;NEED SERVICE, CHECK BRAKES&quot;</td>
</tr>
<tr>
<td>&quot;04 DRIVER'S REPORT&quot;</td>
<td>&quot;HYD CAP CHECK ENGINE LIGHT ON&quot;</td>
</tr>
<tr>
<td>&quot;40 NEGLIGENCE&quot;</td>
<td>&quot;TARP VALVE STICKINGRIGHT SIDE MIRROR BRACKET BROKEN&quot;</td>
</tr>
<tr>
<td>&quot;13 SNOW BREAKDOWN&quot;</td>
<td>&quot;HANDLES IN CAB LOOSE&quot;</td>
</tr>
<tr>
<td>&quot;04 DRIVER'S REPORT&quot;</td>
<td>&quot;NO PLOW LIGHTS&quot;</td>
</tr>
<tr>
<td>&quot;10 ROADCALL&quot;</td>
<td>&quot;WILL NOT START&quot;</td>
</tr>
<tr>
<td>&quot;10 ROADCALL&quot;</td>
<td>&quot;WILL NOT START&quot;</td>
</tr>
<tr>
<td>&quot;10 ROADCALL&quot;</td>
<td>&quot;WILL NOT START&quot;</td>
</tr>
<tr>
<td>&quot;10 ROADCALL&quot;</td>
<td>&quot;WILL NOT START&quot;</td>
</tr>
<tr>
<td>&quot;10 ROADCALL&quot;</td>
<td>&quot;WILL NOT START&quot;</td>
</tr>
<tr>
<td>&quot;10 ROADCALL&quot;</td>
<td>&quot;WILL NOT START&quot;</td>
</tr>
<tr>
<td>&quot;04 DRIVER'S REPORT&quot;</td>
<td>&quot;CONVEYOR NOT WORKING&quot;</td>
</tr>
<tr>
<td>&quot;10 ROADCALL&quot;</td>
<td>&quot;DON'T START&quot;</td>
</tr>
<tr>
<td>&quot;10 ROADCALL&quot;</td>
<td>&quot;DON'T START&quot;</td>
</tr>
<tr>
<td>&quot;10 ROADCALL&quot;</td>
<td>&quot;DON'T START&quot;</td>
</tr>
</tbody>
</table>
Working with Text Data
Working with Text Data

Text Analytics Toolbox
MATLAB

© 2019 The MathWorks, Inc.
Creating Your Own Data
Identifying the Useful Data

1. Acquire Data
2. Preprocess Data
3. Identify Condition Indicators
4. Train Model
5. Machine Learning
6. Deploy & Integrate

- Visualize data
- Extract Features
- Select the most useful features
Identifying the Useful Data

Predictive Maintenance Toolbox
Identifying the Useful Data

- **Signal Features**
  - Generate statistics from signals

- **Rotating Machinery Features**
  - Generate features from rotating machinery signals

- **Nonlinear Features**
  - Generate nonlinear features from signals

**Spectral Features**

Condition variable: `faultCode`

- **Computation mode**: use full signal

- **Spectral peaks**
  - Peak amplitude
  - Peak frequency

- **Peak value lower threshold**: `Inf`
- **Number of peaks**: 1
- **Minimum frequency gap**: 0.001
- **Peak excursion tolerance**: 0

- **Modal coefficients**

- **Band power**
Identifying the Useful Data
Find out more: Manutenzione Predittiva con MATLAB

Francesco Alderisio
Traccia A – 14:30

Data Science e Predictive Analytics

Francesco Alderisio
Postazione Demo
Designing Decision Logic with Stateflow

```matlab
inNormalRegion = true;
counter = 0;
for i=1:length(inData)
    if(inNormalRegion)
        if(inData(i)<t1)
            counter = counter+1;
            if(counter>=N1)
                inNormalRegion = false;
            end
        else
            counter = 0;
        end
    else
        if(inData(i)>=t2)
            counter = counter+1;
            if(counter>=N2)
                inNormalRegion = true;
            end
        else
            counter = 0;
        end
    end
    if(inNormalRegion)
        outData(i) = inData(i);
    else
        outData(i) = 0;
    end
end
```

Diagram:
- **Normal**
  - \( \text{count}(u<t1) \geq N1 \)
  - \( y=u \)
- **Abnormal**
  - \( \text{count}(u \geq t2) \geq N2 \)
  - \( y=0 \)
Using Stateflow in MATLAB

 Callbacks that handle component events

 % Code that executes after component creation
 function startupFcn(app)
     app.lanternlogic = blink.lanternLogic('app',app);
 end

 % Button pushed function: POWERButton
 function POWERButtonPushed(app, event)
     app.lanternLogic.powerButton();
 end

 % Button pushed function: COLORButton
 function COLORButtonPushed(app, event)
     app.lanternLogic.colorButton();
 end

 % Close request function: UIFigure
 function UIFigureCloseRequest(app, event)
     delete(app.lanternLogic);
     delete(app);
 end

 % Button pushed function: BLINKButton
 function BLINKButtonPushed(app, event)
     app.lanternLogic.blinkButton();
 end
Editing at the Speed of Thought
Editing at the Speed of Thought
Editing at the Speed of Thought
Editing at the Speed of Thought
Controlling the Execution of Model Components

Schedulable Rate-Based Model

Export Function Model
Controlling the Execution of Model Components
Simplifying Integration with External C/C++ Code

Simulink Coder

Column-Major

#include "rtwdemo_row2col2row_workflow_row2row.h"

/* Block parameters (default storage) */

PrtP = {
    /* Variable: Tbl_1 */
    /* Referenced by: '(Root)/2-D Lookup Table' */
    { 1.0, 2.0, 3.0, 4.0, 5.0, 6.0, 7.0, 8.0, 9.0, 10.0 },
    { 11.0, 12.0, 13.0, 14.0, 15.0, 16.0, 17.0, 18.0, 19.0, 20.0, 21.0, 22.0, 23.0, 24.0, 25.0, 26.0, 27.0, 28.0, 29.0, 30.0, 31.0, 32.0, 33.0, 34.0, 35.0, 36.0, 37.0, 38.0, 39.0, 40.0, 41.0, 42.0, 43.0, 44.0, 45.0, 46.0, 47.0, 48.0, 49.0, 50.0, 51.0, 52.0, 53.0, 54.0, 55.0, 56.0, 57.0, 58.0, 59.0, 60.0 };
};
Simplifying Integration with External C/C++ Code

Row-Major
Viewing Generated Code Alongside the Model

Fuel Rate Control Subsystem

- Module inputs:
  - single D2 (g/s)
  - <speed>
- Module outputs:
  - single D2 (g/s)
  - est_airflow (g/s)
- Internal signals:
  - sensors
  - C2_normal
  - es_mode
  - airflow_calc
  - fuel_mode
  - fb_correction
  - fuel_rate
  - estimate_airflow (g/s)
  - single D2 (g/s)
- Control logic:
  - validate_sample_time
  - boolean D2
  - C2_normal
  - es_i
  - EngSensors D2
Viewing Generated Code Alongside the Model
Sharing Live Scripts

Estimating Sunrise and Sunset

Using the latitude ($\phi$), the sun's declination ($\delta$) and the solar time correction ($SC$) we can calculate sunrise and sunset times.

\[
\text{sunrise} = 12 - \frac{\cos^{-1}(-\tan \phi \tan \delta)}{15^\circ} - \frac{SC}{60}
\]

\[
\text{sunset} = 12 + \frac{\cos^{-1}(-\tan \phi \tan \delta)}{15^\circ}
\]

Refer to this page for background and details on the equations used.
Sharing Live Scripts

Exploring Exoplanets

In this example we will explore some data on exoplanets - planets outside our own solar system. The data we use here is a subset of data from the NASA Exoplanet Archive. We will start by using the data to answer some questions about the set of exoplanets in the archive. Then we will do some calculations to try to identify planets in the archive that might be capable of supporting life.

exoplanets = readtable('exoplanets.csv');
exoplanets sizel);

How Far Away Are these Planets?

There are 90 exoplanets within 50 light-years of earth and 460 exoplanets within 200 light-years.

histogram(x26iexoplanets.St_Distance, 'BinWidth', 50)
xlabel('Number of Planets')
 ylabel('Light Years from Earth')

Where is the nearest exoplanet?

lid = find(exoplanets.St_Distance == min(exoplanets.St_Distance))
name = char(exoplanets.St_Name(lid, 1:

MATLAB
Sharing Live Scripts

![Live Editor with code and output](image)

- **P**: 1:40
- **Slider**: 350
- **Drop down**: "carbon dioxide"

Graph: "carbon dioxide @ 350 Kelvin"
Creating Apps

Plate Browser Summary Tables

Select Files Current File: microriier_data0001.csv

Microplate Plot

EC50 Curves

<table>
<thead>
<tr>
<th>File</th>
<th>Compound Nr</th>
<th>NegControl</th>
<th>Conc1</th>
<th>Conc2</th>
<th>Conc3</th>
<th>Conc4</th>
<th>Conc5</th>
<th>Conc6</th>
<th>Conc7</th>
<th>Conc8</th>
</tr>
</thead>
<tbody>
<tr>
<td>microriier_d...</td>
<td>1</td>
<td>-0.0741</td>
<td>0.3564</td>
<td>9.8759</td>
<td>56.8743</td>
<td>91.7323</td>
<td>96.7684</td>
<td>97.1532</td>
<td>97.1910</td>
<td>97.1940</td>
</tr>
<tr>
<td>microriier_d...</td>
<td>2</td>
<td>-0.0143</td>
<td>-0.5044</td>
<td>-0.5044</td>
<td>-0.5044</td>
<td>-0.5044</td>
<td>-0.5044</td>
<td>-0.5044</td>
<td>17.0436</td>
<td></td>
</tr>
<tr>
<td>microriier_d...</td>
<td>3</td>
<td>0.0054</td>
<td>-0.4702</td>
<td>3.1908</td>
<td>52.9906</td>
<td>97.5746</td>
<td>100.5606</td>
<td>100.6066</td>
<td>100.6066</td>
<td>100.6066</td>
</tr>
<tr>
<td>microriier_d...</td>
<td>4</td>
<td>0.1096</td>
<td>0.2325</td>
<td>0.2385</td>
<td>0.3712</td>
<td>3.2339</td>
<td>41.1660</td>
<td>94.7343</td>
<td>100.6591</td>
<td>100.9487</td>
</tr>
<tr>
<td>microriier_d...</td>
<td>5</td>
<td>-0.0572</td>
<td>0.7461</td>
<td>1.7104</td>
<td>26.8872</td>
<td>84.5134</td>
<td>96.2335</td>
<td>100.4717</td>
<td>100.5601</td>
<td>100.5700</td>
</tr>
</tbody>
</table>
Deploying Web Apps

MATLAB Web Apps

Transient Conduction

Initial and Boundary Conditions
- Initial T (°C): 10
- Top T (°C): 0
- Bottom T (°C): 50
- Left T (°C): 25
- Right T (°C): 25

Geometry
- x (m): 0.05
- y (m): 0.05
- dx (m): 0.0025
- dy (m): 0.0025

Thermal Diffusivity
- Alpha (m²/s): 1e-4

Time and Convergence
- dt (s): 0.01
- Total Time (s): 50
- Convergence Criterion: 1e-4

Note: Numerical stability requires F
Current Po = 0.0003

Time = 33 s

MATLAB Compiler
Using MATLAB & Simulink to Build Algorithms in Everything

Inputs

Design

Outputs
Evaluating Architectures

Architecture

Inputs

Design

Outputs

MATLAB® & SIMULINK®
Evaluating Architectures

Inputs → Architecture → Design → Outputs

MATLAB® & SIMULINK®
Designing System and Software Architectures
Designing System and Software Architectures
Find out more:
Ingegneria dei sistemi: dai requisiti all’architettura alla simulazione

Vincenzo Petrella
Traccia B – 14:30
Designing **Beyond** System and Software Architectures

**Systems and Software**

System Composer

**SoC Hardware and Software**

SoC Blockset

**AUTOSAR Software**

AUTOSAR Blockset
Using MATLAB & Simulink to Build Algorithms in Everything

Inputs → Architecture → Design → Outputs

Test & Verification → Collaboration → Scaling
Using MATLAB & Simulink to Build Algorithms in Everything
Integrating with Third-party Requirements Tools

External Requirements
- doc
- xls
- Database

Simulink Requirements
- External Requirements
- Authored Requirements

ReqIF
- Import
- Edit
- Export

R2019a
Find out more:
Master Class:
Sviluppo di un sistema di gestione delle batterie con Simulink
Traccia B - 15:30
Maurizio Dalbard – Aldo Caraceto

Sviluppo Model-Based Design applicato ad un sistema di gestione delle batterie
Postazione Demo
Maurizio Dalbard – Vincenzo Petrella
Find out more:
Sviluppare controlli digitali per convertitori elettronici di potenza
Traccia B - 13:30
Aldo Caraceto

Industry 4.0: simulazione dinamica closed-loop e test virtuale
Postazione Demo
Aldo Caraceto
Using the MATLAB Unit Test Framework

```
>> result.table
ans =
    2×6 table

         Name                      Passed  Failed  Incomplete  Duration  Details
    ________________________  _______  _______  __________    _______  _______
  'test_Predictions/Test_ModelType'    true    false       false    0.12241    [1×1 struct]
  'test_Predictions/Test_Prediction'    false     true       true    0.11542    [1×1 struct]
```
Using the MATLAB App Testing Framework

testCase.press(myApp.checkbox)

testCase.choose(myApp.discreteKnob, "Medium")

testCase.drag(myApp.continuousKnob, 10, 90)

testCase.type(myApp.editfield, myTextVar)
Using the MATLAB Performance Testing Framework
Using Continuous Integration

Plugins Index

Discover the 1000+ community contributed Jenkins plugins to support building, deploying and automating any project.

Browse categories
- Platforms
- User interface
- Administration
- Source code management

New Plugins
- ORebel
- MATLAB
- MISRA Compliance Report
- Zoom
- VectorCAST Execution
- Klocwork Community
- JQuery
- Analysis Model API
Using Continuous Integration

The Jenkins plugin for MATLAB® enables you to easily run your MATLAB tests and generate test artifacts in formats such as JUnit, TAP, and Cobertura code coverage reports.

Features
- Support to run MATLAB tests, present in the Jenkins workspace automatically. (This also includes the tests present in .prj files)
- Generate tests artifacts in JUnit, TAP & Cobertura code coverage formats.
- Support to run tests, using custom MATLAB command or custom MATLAB script file.
Using Projects in MATLAB

![MATLAB Project Management Interface]

- **Project (226)**
- **Modified (344)**
- **Classification**
  - Test
  - Design
Using Projects in MATLAB

Find out more:
Master Class: Sviluppo software con MATLAB

Traccia A – 15:30
Francesco Alderisio - Giuseppe Ridinò
Parallel Simulations in Simulink

Simulation Manager

batchsim

Simulink
Parallel Computing Toolbox
Scaling Computations on Clusters and Clouds

MATLAB

Parallel Computing Toolbox

MATLAB Parallel Server

Cloud

GPU

Multi-core CPU
Using MATLAB & Simulink to Build Algorithms in Everything

Inputs → Architecture → Design → Outputs

- Test & Verification
- Collaboration
- Scaling

MATLAB & SIMULINK®
Specialized Tools for Building Algorithms in Everything

Communications

Physical interconnects

Analog Mixed-Signal

- 5G Toolbox
- SerDes Toolbox
- Mixed-Signal Blockset
Developing Autonomous Systems

Perception

Planning

Control
Evaluate Sensor Fusion Architectures
Simulate Path Planning Algorithms
Design Lane-following and Spacing Control Algorithms
Developing Autonomous Systems

- Lidar Processing & Tracking
  - Computer Vision Toolbox

- HERE HD Maps & OpenDRIVE Roads
  - Automated Driving Toolbox

- UAV Algorithms
  - Robotics System Toolbox
Using MATLAB & Simulink to Build Algorithms in Everything

Inputs → Architecture → Design → Outputs

Test & Verification → Collaboration → Scaling
Read the Release Notes

R2019a at a Glance

Explore What's New
Get more out of MATLAB and Simulink by downloading the latest release.

Download release now

Release Highlights

Deep Learning
Develop controllers and decision making systems using reinforcement learning, train deep learning models on NVIDIA DGX and cloud platforms, and apply deep learning to 3-D data.

» Learn more

Automotive
Design and simulate AUTOSAR software, interface with HERE HD maps, and generate energy balance reports.

» Learn more

Systems Engineering
Design and analyze system and software architectures with System Composer.

» Learn more

Projects
Use projects in MATLAB and Simulink to organize, manage, and share your work.

» Learn more

© 2019 The MathWorks, Inc.

86
Get Started

**MATLAB Onramp**
Quickly learn the essentials of MATLAB.

**Simulink Onramp**
Learn to create, edit, and troubleshoot Simulink models.

**Deep Learning Onramp**
Learn to use deep learning techniques in MATLAB for image recognition.
## Attend Sessions this Afternoon

<table>
<thead>
<tr>
<th>Time</th>
<th>Traccia A</th>
<th>Traccia B</th>
</tr>
</thead>
<tbody>
<tr>
<td>12:00</td>
<td>Pranzo, Tech Talks e area espositiva</td>
<td></td>
</tr>
</tbody>
</table>
| 13:30 | **Deep Learning e Reinforcement Learning per l’intelligenza artificiale**  
*Giuseppe Ridinò, MathWorks*  
**Sviluppare controlli digitali per convertitori elettronici di potenza**  
*Aldo Caraceto, MathWorks*                                                                 |                                                                                                   |
| 14:00 | **Dagli script a linguaggio di programmazione: una GUI per la produzione**  
*Marco Basilico, TRE ALTAMIRA*  
**Sviluppo di un sistema di sospensioni semiattive mediante Model-Based Design con architettura AUTOSAR e conforme allo standard A-SPICE**  
*Andrea Palazzetti, Magneti Marelli*                                                                 |                                                                                                   |
| 14:30 | **Manutenzione Predittiva con MATLAB**  
*Francesco Alderisio, MathWorks*  
**Ingegneria dei sistemi: dai requisiti all’architettura alla simulazione**  
*Vincenzo Petrella, MathWorks*                                                                 |                                                                                                   |
| 15:00 | **Pausa caffè e area espositiva**                                                                                                          |                                                                                                   |
| 15:30 | **Master Class: Sviluppo software con MATLAB**  
*Francesca Perino e Giuseppe Ridinò, MathWorks*  
**Master Class: Sviluppo di un sistema di gestione delle batterie con Simulink**  
*Aldo Caraceto e Maurizio Dalbard, MathWorks*                                                                 |                                                                                                   |
| 17:00 | **Chiusura lavori**                                                                                                                          |                                                                                                   |